Collect examples (pairs of tasks in sequence) — Step 1

↓

Form distinct example clusters — Step 2

↓

Use clusters to produce probability distributions for later use — Step 3

For each cluster:

↓

Split learning problem into a set of problems, one per example attribute — Step 4

↓

Perform specialised learning on each attribute — Step 5

↓

Combine rules collected, filter for redundancy — Step 6

↓

Add results for each cluster to model — Step 7

↓

END

Figure 1. Process Diagram for Construction of the User Model

```
                                                            ─ Step 20
  ┌────────────────────────────────┐   ┌──╯
  │  Collect initial task from user│ ◄─┘
  └────────────────────────────────┘
                  │
                  ▼
  ┌────────────────────────────────┐
  │ Consult model for possible     │
  │ tasks immediately in sequence, │ ◄──── Step 21
  │ using probability distributions│
  │ to filter out unlikely answers.│
  └────────────────────────────────┘
                  │
                  ▼
  ┌────────────────────────────────┐
  │ Build tree of possible sequences,│
  │ consulting model at each level for│
  │ tasks which could go in sequence │ ◄──── Step 22
  │ and using probability distributions│
  │ to filter out unlikely answers. │
  └────────────────────────────────┘
                  │
                  ▼
  ┌────────────────────────────────┐
  │ Present sequence suggestions   │ ◄──── Step 24
  │ to user.                       │
  └────────────────────────────────┘
                  │
                  ▼
              ┌───────┐
              │  END  │
              └───────┘
```
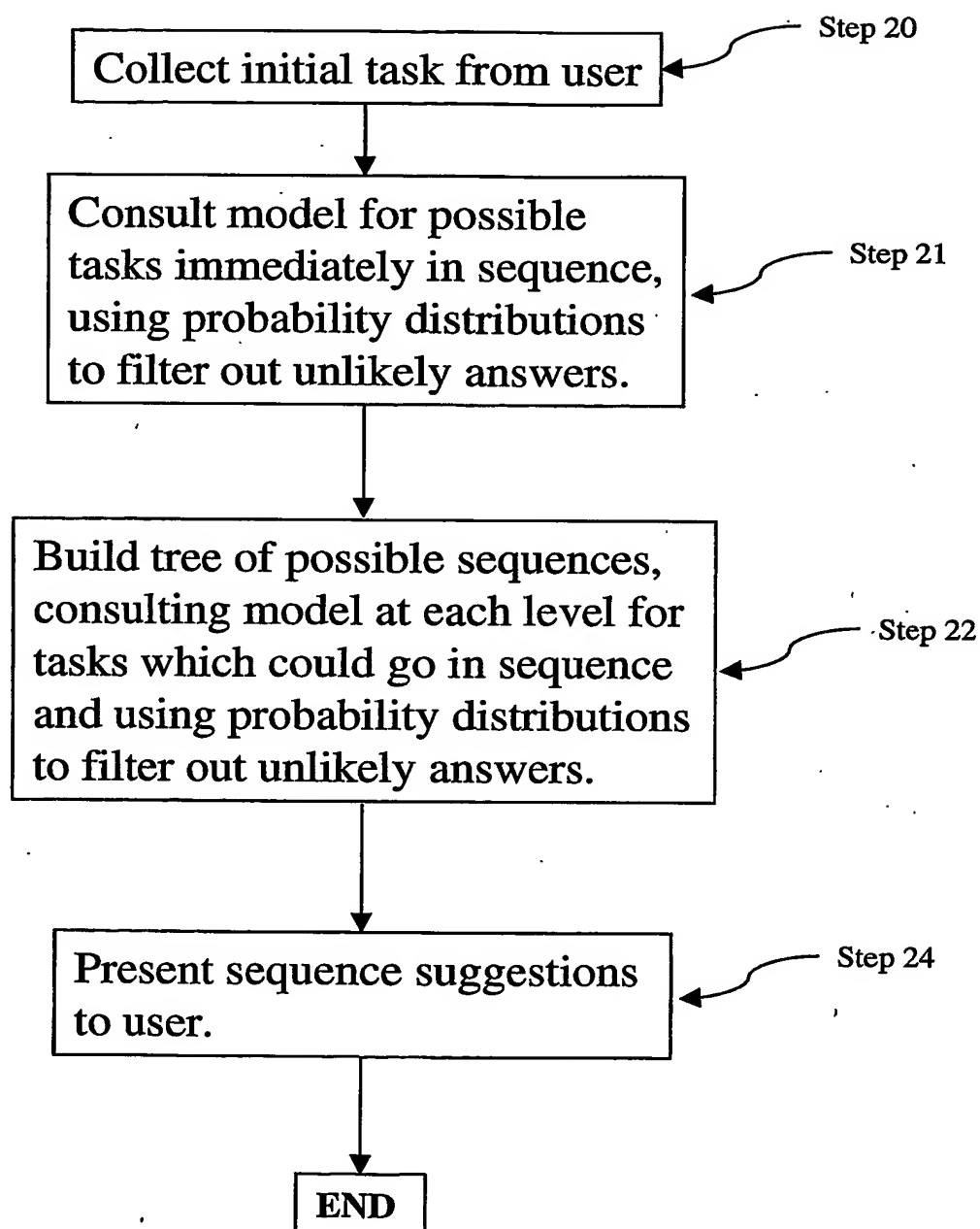
Figure 2a: Summary Diagram showing Prediction of Sequences of Tasks

3/5

20 — **Get query**

21A —   — 21B

-     -
| Generate sequence of following tasks: | | Generate sequence of preceding tasks: |
-     -

210 — **Make task into root node of tree of possible answers**   — 22

- -
| Construct tree (Steps 221 to 228): |
- -

Step 221: Pick an unexpanded node

Step 222: Find all tasks which could form possible sub-nodes

Step 223: No. of answers returned > 0? — Yes

No

Step 224: Bucket sort answers using stereotypes from probability distributions as categories

Step 225: Use probability distributions to give each answer a ranking score

Step 226: Choose from each bucket the answer with the highest score

Step 227: Answers form new sub-nodes of picked node in tree

Step 228: More unexpanded nodes?

Yes

No

Parse tree to extract possible sequences of tasks —— 23

Return answers to be presented to the user —— 24
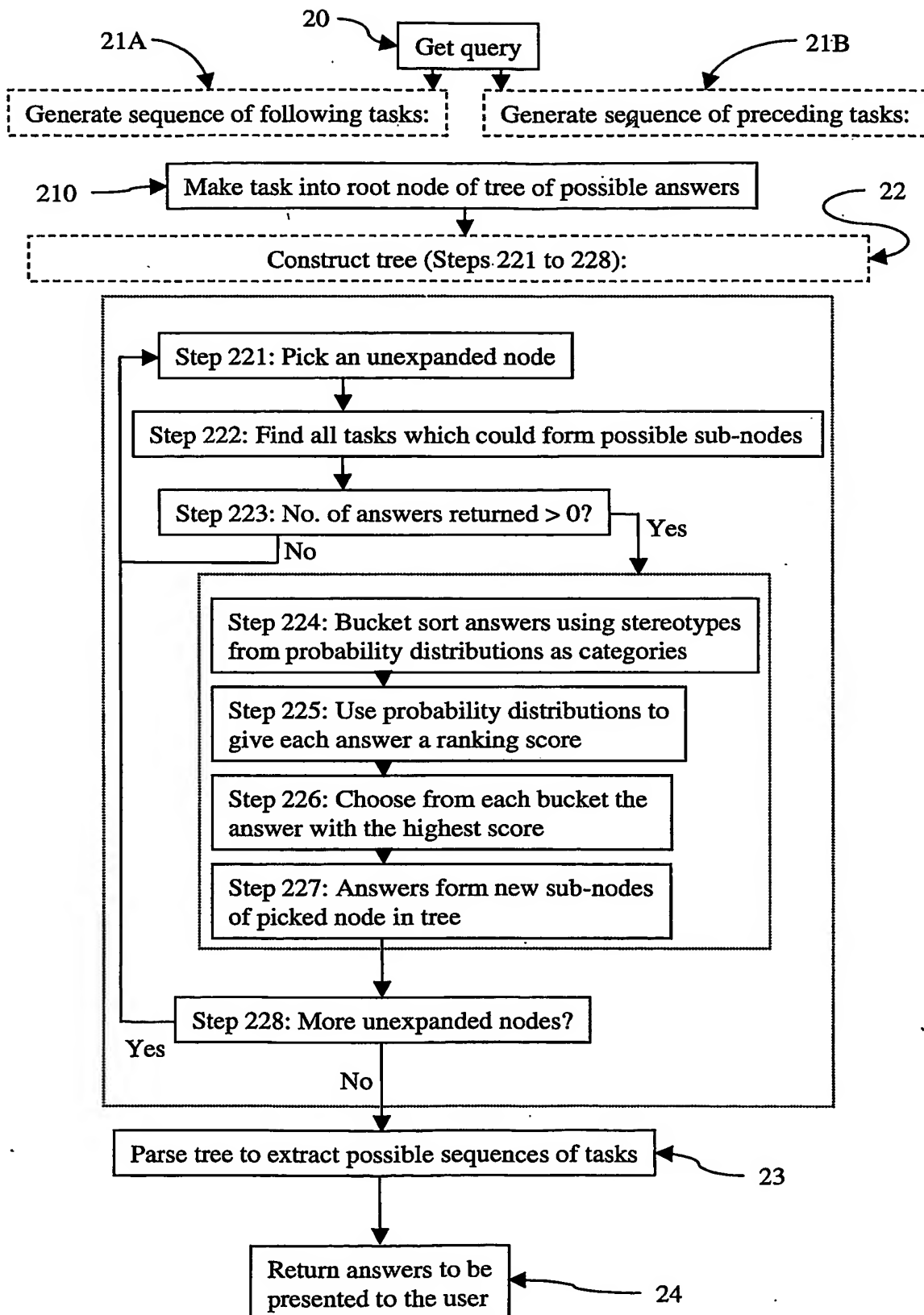
Figure 2b: Process Diagram showing Prediction of Sequences of Tasks

301: Get query tasks

For each task: 302
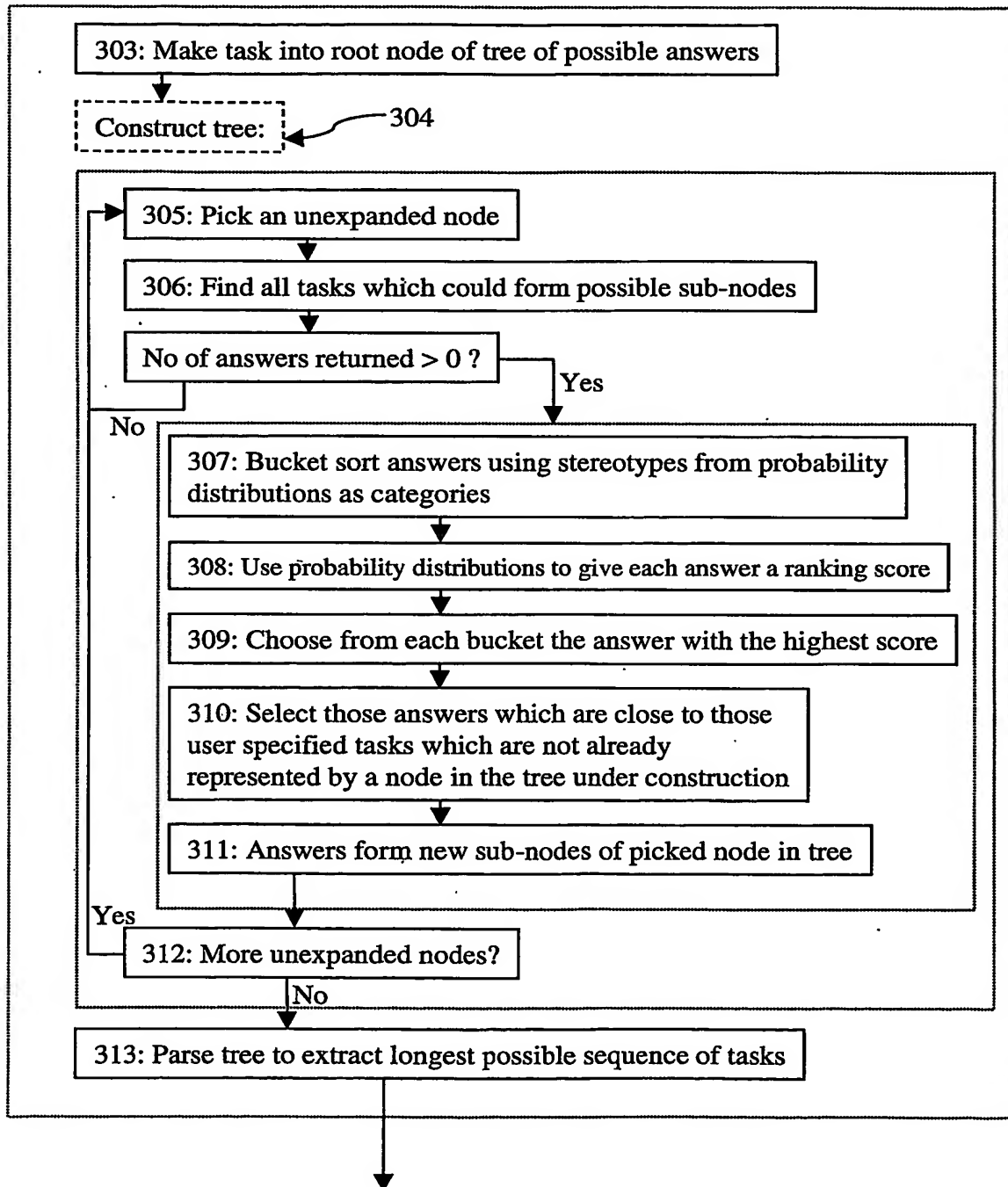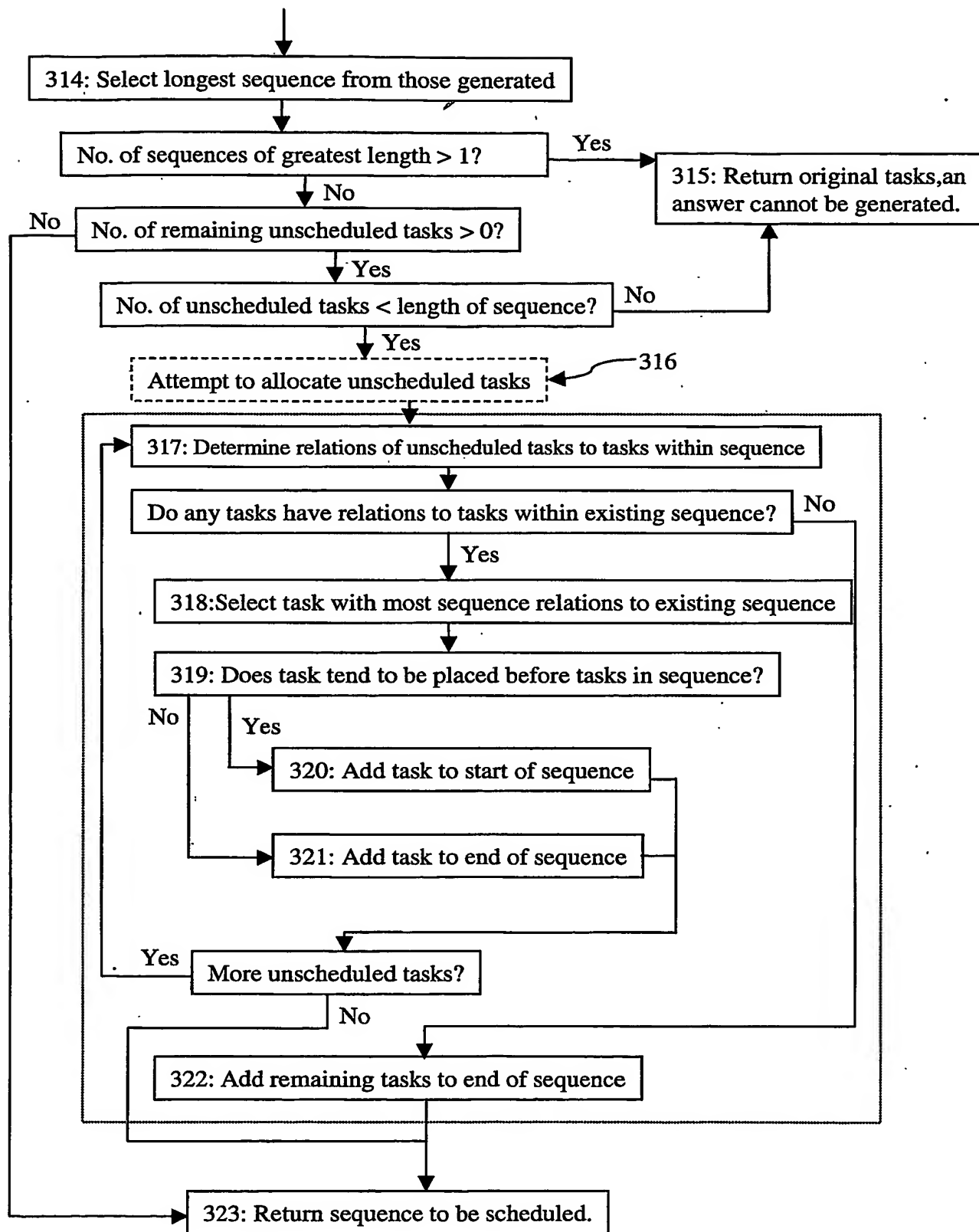Generate sequence of following tasks:

303: Make task into root node of tree of possible answers

Construct tree: 304

305: Pick an unexpanded node

306: Find all tasks which could form possible sub-nodes

No of answers returned > 0 ?

Yes

No

307: Bucket sort answers using stereotypes from probability distributions as categories

308: Use probability distributions to give each answer a ranking score

309: Choose from each bucket the answer with the highest score

310: Select those answers which are close to those user specified tasks which are not already represented by a node in the tree under construction

311: Answers form new sub-nodes of picked node in tree

Yes

312: More unexpanded nodes?

No

313: Parse tree to extract longest possible sequence of tasks

Figure 3 (continued)

Figure 3: Process Diagram for Predicting Sequences for Sets of Tasks

5/5

314: Select longest sequence from those generated

No. of sequences of greatest length > 1? ——Yes→ 315: Return original tasks, an answer cannot be generated.

No

No. of remaining unscheduled tasks > 0?

No

Yes

No. of unscheduled tasks < length of sequence? ——No→

Yes

Attempt to allocate unscheduled tasks ◄——316

317: Determine relations of unscheduled tasks to tasks within sequence

Do any tasks have relations to tasks within existing sequence? ——No→

Yes

318: Select task with most sequence relations to existing sequence

319: Does task tend to be placed before tasks in sequence?

No

Yes

320: Add task to start of sequence

321: Add task to end of sequence

Yes

More unscheduled tasks?

No

322: Add remaining tasks to end of sequence

323: Return sequence to be scheduled.

Figure 3 (continued)